

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

31000 U.S. PRO  
10/075989  
02/16/02

別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
with this Office

出 願 年 月 日

Date of Application:

2001年 2月20日

出 願 番 号

Application Number:

特願2001-043487

ST.10/C]:

[JP2001-043487]

願 人

Applicant(s):

セイコーエプソン株式会社

U.S. Appln Filed 2-15-02  
Inventor: M. Oyanagi  
Mattingly Stanger & Malor  
Docket NGB-103

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2002年 1月25日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



出証番号 出証特2002-3000659

【書類名】 特許願

【整理番号】 12966801

【提出日】 平成13年 2月20日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 3/00

【発明の名称】 マルチファンクションプリンタ

【請求項の数】 9

【発明者】

【住所又は居所】 長野県諏訪市大和三丁目3番5号 セイコーエプソン株式会社内

【氏名】 小 柳 誠

【特許出願人】

【識別番号】 000002369

【住所又は居所】 東京都新宿区西新宿二丁目4番1号

【氏名又は名称】 セイコーエプソン株式会社

【代理人】

【識別番号】 100075812

【弁理士】

【氏名又は名称】 吉 武 賢 次

【選任した代理人】

【識別番号】 100088889

【弁理士】

【氏名又は名称】 橋 谷 英 俊

【選任した代理人】

【識別番号】 100082991

【弁理士】

【氏名又は名称】 佐 藤 泰 和

【選任した代理人】

【識別番号】 100096921

【弁理士】

【氏名又は名称】 吉 元 弘

【選任した代理人】

【識別番号】 100103263

【弁理士】

【氏名又は名称】 川 崎 康

【選任した代理人】

【識別番号】 100107582

【弁理士】

【氏名又は名称】 関 根 毅

【手数料の表示】

【予納台帳番号】 087654

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 マルチファンクションプリンタ

【特許請求の範囲】

【請求項1】

スキャナ部とプリンタ部とを有するマルチファンクションプリンタであって、コンピュータから、前記プリンタ部をリセットするプリンタ部リセットコマンドを受信する、受信手段と、

前記プリンタ部リセットコマンドを受信した場合に、前記スキャナ部が動作中であっても、前記プリンタ部のみをリセットする、リセット手段と、  
を備えることを特徴とするマルチファンクションプリンタ。

【請求項2】

前記マルチファンクションプリンタは、中央処理装置を1つだけ備えていることを特徴とする請求項1に記載のマルチファンクションプリンタ。

【請求項3】

前記リセット手段は、少なくとも、

前記プリンタ部リセットコマンドを受信した場合には、コンピュータから送信された印刷データを格納する入力バッファに、印刷データとしては無意味なデータと解釈されるダミーデータを印刷データとして書き込む、第1タスクと、

前記入力バッファに格納されている前記印刷データを読み出して、イメージデータに展開する、第2タスクとを、

前記中央処理装置で実行することにより実現される、ことを特徴とする請求項2に記載のマルチファンクションプリンタ。

【請求項4】

前記リセット手段は、前記中央処理装置で実行されるタスクとして、

前記プリンタ部リセットコマンドを受信した場合に、前記プリンタ部がリセット処理中であることを示すリセット中識別情報をセットする、第3タスクをさらに備えることを特徴とする請求項3に記載のマルチファンクションプリンタ。

【請求項5】

前記第2タスクは、最も上位のメインループ処理と、このメインループ処理か

ら階層的に派生する下位のループ処理とを有しており、前記下位のループ処理は、前記リセット中識別情報がセットされている場合には、その処理を抜けて上位のループ処理に戻るよう構成されている、ことを特徴とする請求項 4 に記載のマルチファンクションプリンタ。

【請求項 6】

前記第 2 タスクでは、その時点のモードがキャラクタデータを印刷するキャラクタモードである場合には、前記ダミーデータが所定回数以上連続したかどうかを判断する連続回数判断処理を行い、その時点のモードがグラフィックデータを印刷するグラフィックモードである場合には、この連続回数判断処理を行わない、ことを特徴とする請求項 5 に記載のマルチファンクションプリンタ。

【請求項 7】

スキャナ部とプリンタ部とを有するマルチファンクションプリンタであって、コンピュータから、前記プリンタ部をリセットするプリンタ部リセットコマンドを受信する、受信工程と、

前記プリンタ部リセットコマンドを受信した場合に、前記スキャナ部が動作中であっても、前記プリンタ部のみをリセットする、リセット工程と、

を備えることを特徴とするマルチファンクションプリンタの制御方法。

【請求項 8】

スキャナ部とプリンタ部とを有するマルチファンクションプリンタに、

コンピュータから、前記プリンタ部をリセットするプリンタ部リセットコマンドを受信する、受信ステップと、

前記プリンタ部リセットコマンドを受信した場合に、前記スキャナ部が動作中であっても、前記プリンタ部のみをリセットする、リセットステップと、

を実行させるためのプログラム。

【請求項 9】

スキャナ部とプリンタ部とを有するマルチファンクションプリンタに、

コンピュータから、前記プリンタ部をリセットするプリンタ部リセットコマンドを受信する、受信ステップと、

前記プリンタ部リセットコマンドを受信した場合に、前記スキャナ部が動作中

であっても、前記プリンタ部のみをリセットする、リセットステップと、  
を実行させるためのプログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、スキャナ部とプリンタ部とを有するマルチファンクションプリンタ  
に関し、特に、スキャナ部とプリンタ部とが独立に動作可能なマルチファンク  
ションプリンタに関する。

【0002】

【従来技術】

スキャナ部とプリンタ部とが一体化されて、1つの筐体に格納されたマルチ  
ファンクションプリンタが普及してきている。このようなマルチファンクション  
プリンタにおいては、1台で、スキャナとしての役割と、プリンタとしての役割と  
、コピー機としての役割とを、果たすことができる。この場合、カラー印刷やカ  
ラーコピーを可能にするため、プリンタ部としては、いわゆるカラーインクジェ  
ットプリンタが用いられている場合が多い。

【0003】

このようなマルチファンクションプリンタにおいて、ユーザが何らかの理由に  
より、プリンタ部で印刷中のジョブをキャンセルしたい場合がある。この場合、  
ユーザは、マルチファンクションプリンタに接続されたコンピュータから、リセ  
ットコマンドを送信し、マルチファンクションプリンタでは、このリセットコマ  
ンドに基づいて、印刷中のジョブのキャンセルを行う。

【0004】

【発明が解決しようとする課題】

しかしながら、従来マルチファンクションプリンタにおいては、このキャン  
セルにあたって、印刷を実行するために動作している各種のタスクを初期状態に  
戻すために、リブートの処理を実行していた。すなわち、マルチファンクション  
プリンタ上の各種タスクは、リアルタイムOS (Operating System) 上で動作し  
ているため、このリアルタイムOSそのものをリブートすることにより、マルチ

ファンクションプリンタを初期化して、印刷中のジョブをキャンセルしていた。

【0005】

しかし、スキャナ部が、プリンタ部と独立して動作可能なマルチファンクションプリンタもある。このようなマルチファンクションプリンタにおいては、プリンタ部がコンピュータから受信した印刷データの印刷をしつつ、これと並行して、スキャナ部が原稿等のスキャン動作を行うことが可能になる。このように並行動作している最中に、リアルタイムOSそのものをリブートしてしまうと、本来はユーザがリセットするつもりのないスキャナ部までもが、リセットされてしまう。

【0006】

しかも、ユーザが再度スキャンをやり直そうとして、コマンドをコンピュータからマルチファンクションプリンタに送信すると、リアルタイムOSがリブートして立ち上がり直されているので、通信エラーを起こしてしまう場合も生じてしまう。この場合、ユーザは、再度、コンピュータのスキャナ用のドライバを立ち上げ直して、通信エラーを解消しなければならない。

【0007】

そこで本発明は、前記課題に鑑みてなされたものであり、プリンタ部とスキャナ部とを有するマルチファンクションプリンタにおいて、プリンタ部で行われている印刷動作のみを、キャンセルすることができるようにすることを目的とする。

【0008】

【課題を解決するための手段】

上記課題を解決するため、本発明に係るマルチファンクションプリンタは、スキャナ部とプリンタ部とを有するマルチファンクションプリンタであって、コンピュータから、前記プリンタ部をリセットするプリンタ部リセットコマンドを受信する、受信手段と、前記プリンタ部リセットコマンドを受信した場合に、前記スキャナ部が動作中であっても、前記プリンタ部のみをリセットする、リセット手段と、を備えることを特徴とする。これにより、スキャナ部とプリンタ部とが並行して動作している場合にも、プリンタ部のみをリセットすることができるよ

うになる。

【0009】

この場合、前記マルチファンクションプリンタは、中央処理装置を1つだけ備えようにしてもよい。

【0010】

さらに、前記リセット手段は、少なくとも、前記プリンタ部リセットコマンドを受信した場合には、コンピュータから送信された印刷データを格納する入力バッファに、印刷データとしては無意味なデータと解釈されるダミーデータを印刷データとして書き込む、第1タスクと、前記入力バッファに格納されている前記印刷データを読み出して、イメージデータに展開する、第2タスクとを、前記中央処理装置で実行することにより実現されるようにしてもよい。このように、入力バッファにダミーデータを格納しておくことにより、入力バッファから印刷データを読み出してイメージデータに展開する第2タスクは、どのような処理をしていたとしても、いずれ、その展開処理が終わることになる。

【0011】

さらに、前記リセット手段は、前記中央処理装置で実行されるタスクとして、前記プリンタ部リセットコマンドを受信した場合に、前記プリンタ部がリセット処理中であることを示すリセット中識別情報をセットする、第3タスクを備えるようにしてもよい。これにより、このリセット中識別情報をチェックすることにより、各タスクがその時点でプリンタ部がリセット処理中であることを識別できるようになる。

【0012】

また、前記第2タスクは、最も上位のメインループ処理と、このメインループ処理から階層的に派生する下位のループ処理とを有しており、前記下位のループ処理は、前記リセット中識別情報がセットされている場合には、その処理を抜けて上位のループ処理に戻るよう構成してもよい。これにより、プリンタ部がリセット処理中である場合には、第2タスクに含まれている下位のループ処理は、順次、その処理を抜けて上位のループ処理に移行し、やがて、最も上位のメインループ処理に戻るができる。



## 【0013】

さらに、前記第2タスクでは、その時点のモードがキャラクタデータを印刷するキャラクタモードである場合には、前記ダミーデータが所定回数以上連続したかどうかを判断する連続回数判断処理を行い、その時点のモードがグラフィックデータを印刷するグラフィックモードである場合には、この連続回数判断処理を行わないようにしてもよい。これにより、ダミーデータと同じ印刷データが出現しやすいキャラクタモードにおけるスループットの低下を抑止することができる。

## 【0014】

なお、本発明は上述したようなマルチファンクションプリンタを実現するための制御方法としてとらえることもでき、また、そのためのプログラム及びそのプログラムが記録された記録媒体としてとらえることもできる。

## 【0015】

## 【発明の実施の形態】

本発明は、プリンタ部だけをリセットするプリンタ部リセットコマンドを受信したことを、必要なタスクに通知するとともに、印刷処理に関する各種タスクがその時点で行っている処理から抜け出せるように、印刷データを格納する入力バッファに、印刷データの解釈上は意味のないダミーデータを連続して書き込むようにしたものである。そして、これにより、印刷処理に関する各種の複雑なタスクが、その時点で行っている処理から自ら抜け出して、プリンタ部のみをリセットすることができるようにしたものである。以下、本発明に係る一実施形態を詳細に説明する。

## 【0016】

まず、図1に基づいて、本実施形態に係るマルチファンクションプリンタ10の内部構成を説明する。この図1は、スキャナ部とプリンタ部とが一体化されたマルチファンクションプリンタ10の内部構成を示すブロック図である。

## 【0017】

図1に示すように、マルチファンクションプリンタ10は、RAM (Random Access Memory) 12と、ROM (Read Only Memory) 14と、CPU (Central

Processing Unit：中央処理装置）16と、スキャナスキャナASIC（Application Specific IC）18と、プリンタASIC20とを備えており、これらは内部バス22を介して、相互に接続されている。このことから分かるように、本実施形態に係るマルチファンクションプリンタ10においては、CPUは1つだけ設けられている。

## 【0018】

スキャナASIC18には、実際のスキャン動作を行うスキャナ機構部22が接続されている。また、プリンタASIC20には、実際に印刷動作を行うプリントエンジン24が接続されている。

## 【0019】

さらに、マルチファンクションプリンタ10は、USBケーブルを介してコンピュータに接続されるUSB（Universal Serial Bus）ポート26を備えている。このUSBポート26はUSBハブ28に接続されており、USBハブ28はスキャナASIC18とプリンタASIC20とに接続されている。したがって、コンピュータから送信されたデータやコマンドは、USBハブ28を介して、スキャナASIC18とプリンタASIC20の双方に入力される。

## 【0020】

例えば、コンピュータからマルチファンクションプリンタ10に送信された印刷データは、プリンタASIC20で受信され、RAM12に形成された入力バッファに格納される。入力バッファに格納された印刷データは、順次、イメージデータに展開されて、プリンタASIC20を介して、プリントエンジン24で印刷される。本実施形態に係るプリントエンジン24は、カラーのインクジェットヘッドを有しており、このヘッドをキャリッジに搭載して主走査方向に交互に移動することにより、カラー印刷が可能になっている。

## 【0021】

一方、スキャナ機構部10は、光学的に原稿を読み取るラインイメージセンサを有している。このラインイメージセンサは、キャリッジに搭載されており、キャリッジを原稿の一端側から他端側まで移動させることにより、原稿全体を読み取ることが可能である。この読み取り動作は、スキャナASIC12が制御して

おり、読み取ったスキャンデータは、RAM 12に形成されたスキャナバッファに格納される。

#### 【0022】

本実施形態に係るマルチファンクションプリンタ10においては、スキャナ部を構成するスキャナ機構部22におけるスキャン動作と、プリンタ部を構成するプリントエンジン24における印刷動作とが、並行して行えるようになっている。例えば、スキャナ機構部22で原稿のスキャン動作をしながら、プリントエンジン24でコンピュータから送信されてきている印刷データの印刷を、順次、行うことが可能である。また、本実施形態に係るマルチファンクションプリンタ10においては、スキャナ部とプリンタ部とが並行して動作している状態で、プリンタ部のみをリセット（印刷ジョブのキャンセル）することが可能である。

#### 【0023】

このプリンタ部のみをリセットする際の全体的な処理の流れを、図2に基づいて説明する。この図2は、プリンタ部をリセットし、スキャナ部の動作は継続させるためのプリンタ部リセットコマンドを、コンピュータから受信した場合のマルチファンクションプリンタ10の処理を説明するタイミングチャートである。

#### 【0024】

図2に示すように、本実施形態に係るマルチファンクションプリンタ10は、インターフェース制御タスクT10と、制御コマンド解釈タスクT20と、メインタスクT30と、ジョブコントロールタスクT40とを、備えている。なお、これ以外のタスクも、マルチファンクションプリンタ10は備えているが、ここでは、本実施形態に関係のあるタスクのみを示している。また、これらのタスクは、このマルチファンクションプリンタ10の備えるリアルタイムOS上で管理されており、CPU16で実行される。

#### 【0025】

まず、ユーザが印刷中のジョブをキャンセルするために、コンピュータから、プリンタ部リセットコマンドを、マルチファンクションプリンタ10に送信する。このプリンタ部リセットコマンドは、インターフェース制御タスクT10により受信される。インターフェース制御タスクT10は、受信したコマンドが制御

系のコマンドであるか、印刷データ等であるのかを判断し、この例では制御系のコマンドであるので、このプリンタ部リセットコマンドの解釈を要求すべく、これを制御コマンド解釈タスクT20に送信する。

## 【0026】

これを受信した制御コマンド解釈タスクT20が、これを解釈することにより、受信したコマンドがプリンタ部リセットコマンドであることが判明する。このため、制御コマンド解釈タスクT20は、インターフェース制御タスクT10にリプライを依頼し、インターフェース制御タスクT10はこれに基づいてリプライをコンピュータに返信する。また、制御コマンド解釈タスクT20は、リセットコマンド受信メッセージを、ジョブコントロールタスクT40に送信する。

## 【0027】

リセットコマンド受信メッセージを受け取ったジョブコントロールタスクT40は、リセット中フラグRSFをセットするとともに、リセット処理移行要求を、インターフェース制御タスクT10へ送信する。そして、ジョブコントロールタスクT40は、それ以降の印刷、紙送り要求などのジョブを、徹底的に読み捨てる。但し、応答が必要なメッセージについては、とりあえず必要な返信のみ行っておく。

## 【0028】

このリセット処理移行要求を受け取ったインターフェース制御タスクT10は、印刷データを格納しておく入力バッファの全領域に16進数の「00」である0x00をダミーデータとして書き込み、受信データ管理タスクに、データが到着した旨を通知する。

## 【0029】

プリンタ部が印刷中の場合、メインタスクT30では、様々なループ処理を行っているが、印刷データが0x00、つまり、いわゆるNULLであるので、それらのループの処理はやがて終了する。なぜなら、NULLというコマンドは、通常、存在しないからである。

## 【0030】

メインタスクで取得すべき印刷データが、入力バッファになくなった場合は、

受信データ管理タスク（図示省略）を経由して、印刷データ獲得要求がインターフェース制御タスク T 1 0 に発行される。この場合も、インターフェース制御タスク T 1 0 は、0 x 0 0 のダミーの印刷データを入力バッファに書き込み、受信データ管理タスクに、データが到着した旨を通知する。

#### 【 0 0 3 1 】

このように、メインタスク T 3 0 は、N U L L データのみを取得することになるので、いずれ下位のループ処理を抜けて、メインタスク T 3 0 の最上位のループ処理に戻ってくることができる。最上位のループ処理に戻った場合には、メインタスク T 3 0 は、ジョブコントロールタスク T 4 0 に対して、リセット処理終了確認通知を送信し、メインタスク T 3 0 が最上位のループ処理に戻ったことを通知する。

#### 【 0 0 3 2 】

リセット処理終了確認通知を受け取ったジョブコントロールタスク T 4 0 は、リセット中フラグ R S F をリセットし、印刷や紙送り要求などのジョブの読み捨てを中止する。また、リセット処理解除要求を、インターフェース制御タスク T 1 0 へ送信する。これにより、インターフェース制御タスク T 1 0 は、リセット処理モードから、通常の印刷データを受信するモードに戻る。一方、ジョブコントロールタスク T 4 0 は、メインタスク T 3 0 へ、リセット処理終了確認通知に対する返信を行う。この返信がメインタスク T 3 0 に戻ってきた場合、メインタスク T 3 0 は、必要な設定状態を初期状態に戻す。これにより、プリンタ部のみのリセットが行われたことになる。

#### 【 0 0 3 3 】

以上、マルチファンクションプリンタ 1 0 がプリンタ部リセットコマンドを受信した場合の処理を概略的に説明したが、次に、上述した各タスク毎にその詳細な処理内容を説明する。

#### 【 0 0 3 4 】

図 3 は、上述したインターフェース制御タスク T 1 0 で行われるインターフェース制御処理を説明するフローチャートである。本実施形態においては、このインターフェース制御処理は、R O M 1 4 に格納されているインターフェース制御

プログラムを、リアルタイムOS上でCPU16が実行することにより実現される。

#### 【0035】

図3に示すように、インターフェース制御タスクT10は、まず、何からのメッセージを受信したかどうかを判断する（ステップS10）。何らメッセージを受信していない場合（ステップS10：No）には、このステップS10の処理を繰り返して待機する。

#### 【0036】

何らかのメッセージを受信した場合（ステップS10：Yes）には、そのメッセージが、ジョブコントロールタスクT40から送信されたりセット処理移行要求であるかどうかを判断する（ステップS11）。受信したメッセージがリセット処理移行要求である場合（ステップS11：Yes）には、インターフェース制御タスクT10は、入力バッファの全領域に0x00を書き込む（ステップS12）。

#### 【0037】

次に、インターフェース制御タスクT10は、メインタスクT30に、入力バッファに受信データを格納した旨を通知する（ステップS13）。上述したように、この通知は、受信データ管理タスクを介して、メインタスクT30に送信される。

#### 【0038】

次に、インターフェース制御タスクT10は、リセット処理中フラグWSFをセットする（ステップS14）。このリセット処理中フラグWSFは、インターフェース制御タスクT10が、現在、プリンタ部がリセット処理モードにあるか、通常の印刷データを受信するモードにあるのかを、区別するためのフラグである。そして、インターフェース制御タスクT10は、上述したステップS10に戻る。

#### 【0039】

一方、上述したステップS11で、受信したメッセージがリセット処理移行要求でないと判断した場合（ステップS11：No）には、受信したメッセージが

、メインタスクT30からの印刷データの要求であるかどうかを判断する（ステップS15）。上述したように、このメッセージは、メインタスクT30から、受信データ管理タスクを介して、受信される。

## 【0040】

受信したメッセージが印刷データの要求である場合（ステップS15：Yes）には、インターフェース制御タスクT10は、リセット処理中フラグWSFをチェックして、現在、リセット処理中であるかどうかを判断する（ステップS16）。リセット処理中でない場合（ステップS16：No）には、メインタスクT30が印刷データの受信待ちである旨をセットして（ステップS17）、上述したステップS10に戻る。

## 【0041】

一方、ステップS16で、現在、リセット処理中であると判断した場合（ステップS16：Yes）には、入力バッファに0x00を書き込むべく、ステップS12からの処理を実行する。

## 【0042】

これに対して、上述したステップS15で、受信したメッセージが、メインタスクT30からの印刷データの要求でないと判断した場合（ステップS15：No）には、受信したメッセージが、ジョブコントロールタスクT40からのリセット処理解除要求であるかどうかを判断する（ステップS18）。受信したメッセージが、リセット処理解除要求である場合（ステップS18：Yes）には、インターフェース制御タスクT10は、入力バッファへの0x00書き込みを停止する（ステップS19）。続いて、リセット処理中フラグWSFをリセットして（ステップS20）、上述したステップS10に戻る。

## 【0043】

一方、上述したステップS18で、受信したメッセージがリセット処理解除要求でないと判断した場合（ステップS18：No）には、該当するメッセージについて必要な処理を実行する（ステップS21）。例えば、上述したように、コンピュータからのプリンタ部リセットコマンドなどの制御系のコマンドを受信した場合には、これを制御コマンド解釈タスクT20へ転送する。また、例えば、

制御コマンド解釈タスク T 2 0 からプリンタ部リセットコマンドに対するリプライの要求を受信した場合には、そのリプライをコンピュータへ転送する。そして、上述したステップ S 1 0 に戻る。

#### 【 0 0 4 4 】

次に、図 4 に基づいて、制御コマンド解釈タスク T 2 0 について説明する。図 4 は、上述した制御コマンド解釈タスク T 2 0 で行われる制御コマンド解釈処理を説明するフローチャートである。本実施形態においては、この制御コマンド解釈処理は、ROM 1 4 に格納されている制御コマンド解釈プログラムを、リアルタイム OS 上で CPU 1 6 が実行することにより実現される。

#### 【 0 0 4 5 】

図 4 に示すように、制御コマンド解釈タスク T 2 0 は、まず、何からのメッセージを受信したかどうかを判断する（ステップ S 3 0）。何らメッセージを受信していない場合（ステップ S 3 0 : N o）には、このステップ S 3 0 の処理を繰り返して待機する。

#### 【 0 0 4 6 】

何らかのメッセージを受信した場合（ステップ S 3 0 : Y e s）には、そのメッセージが、インターフェース制御タスク T 1 0 を介して、コンピュータから送信されたプリンタ部リセットコマンドであるかどうかを判断する（ステップ S 3 1）。受信したメッセージがプリンタ部リセットコマンドである場合（ステップ S 3 1 : Y e s）には、コンピュータに対する応答データ列を作成し（ステップ S 3 2）、これをインターフェース制御タスク T 1 0 に送信して、リプライを依頼する（ステップ S 3 3）。

#### 【 0 0 4 7 】

次に、制御コマンド解釈タスク T 2 0 は、ジョブコントロールタスク T 4 0 へ、リセットコマンド受信をメッセージとして送信する（ステップ S 3 4）。そして、上述したステップ S 3 0 に戻る。

#### 【 0 0 4 8 】

一方、上述したステップ S 3 1 で、受信したメッセージがプリンタ部リセットコマンドでないと判断した場合（ステップ S 3 1 : N o）には、該当するメッセ



ージについて必要な解釈を行い、必要な処理を実行する（ステップ S 3 5）。例えば、紙送り要求のメッセージである場合は、その解釈を行い、必要なメッセージをジョブコントロールタスク T 4 0 等へ送信する。そして、上述したステップ S 3 0 の処理に戻る。

#### 【 0 0 4 9 】

次に、図 5 及び図 6 に基づいて、メインタスク T 3 0 について説明する。図 5 は、上述したメインタスク T 3 0 で行われるデータ展開制御処理を説明するフローチャートであり、本実施形態におけるメインループ処理を示している。図 6 は、このデータ展開制御処理から派生する下位のループ処理であるグラフィックマネージャ処理を説明するフローチャートである。本実施形態におけるメインタスク T 3 0 には、グラフィックマネージャ処理以外にも、数多くの下位のループ処理が存在するが、ここでは、グラフィックマネージャ処理を代表例として説明する。また、下位のループ処理から派生するさらに下位のループ処理も存在する。つまり、本実施形態においては、メインタスク T 3 0 のループ処理は階層的に構成されている。

#### 【 0 0 5 0 】

なお、図 5 のデータ展開制御処理はキャラクタ系の印刷データを処理するモードであり、図 6 のグラフィックマネージャ処理はグラフィック系の印刷データを処理するモードである。また、本実施形態においては、これらのデータ展開制御処理及びグラフィックマネージャ処理は、ROM 1 4 に格納されているデータ展開制御プログラム及びグラフィックマネージャプログラムを、リアルタイム OS 上で CPU 1 6 が実行することにより実現される。

#### 【 0 0 5 1 】

図 5 に示すように、メインタスク T 3 0 は、まず、入力バッファにデータを受信したかどうかを判断する（ステップ S 4 0）。データを受信していない場合（ステップ S 4 0 : N o）には、このステップ S 4 0 の処理を繰り返して待機する。

#### 【 0 0 5 2 】

入力バッファにデータを受信した場合（ステップ S 4 0 : Y e s）には、メイ

ンタスク T 3 0 は、入力バッファからデータを 1 バイト読み込む（ステップ S 4 1）。次に、この読み込んだデータが、印刷モード移行要求であるかどうかを判断する（ステップ S 4 2）。本実施形態に係るマルチファンクションプリンタ 1 0 は、モードとして、キャラクタ系の印刷データを印刷するキャラクタモードと、グラフィック系の印刷データを印刷するグラフィックモードとを備えており、両者の間は印刷モード移行要求を受信することにより、切り替わるようになっている。

#### 【 0 0 5 3 】

データが印刷モード移行要求でない場合（ステップ S 4 2 : N o）には、取得したデータが 0 x 0 0 であるかどうかを判断する（ステップ S 4 3）。取得したデータが 0 x 0 0 でない場合（ステップ S 4 3 : N o）には、通常の印刷であると考えられるので、キャラクターモードの通常の処理を実行する（ステップ S 4 4）。そして、上述したステップ S 4 0 の処理に戻る。

#### 【 0 0 5 4 】

一方、取得したデータが 0 x 0 0 である場合（ステップ S 4 3 : Y e s）には、この 0 x 0 0 が所定回数以上（例えば、1 0 2 4 回 = 1 K B）、連続したかどうかを判断する（ステップ S 4 5）。ここで、所定回数以上、0 x 0 0 が連続したかどうかのチェックをするのは、キャラクタモードで 1 バイト毎にリセット中フラグ R S F の確認をすると、通常のキャラクタ印刷の際のスループットが低下してしまうので、これを回避するためである。換言すれば、キャラクタ印刷の場合、印刷データとして 0 x 0 0 を受信することも少なからずあり得るからである。但し、多少のスループットの低下を許容して、このような判断をしないようにしてもよい。

#### 【 0 0 5 5 】

したがって、0 x 0 0 が所定回数以上連続していない場合（ステップ S 4 5 : N o）には、上述した通常のキャラクタモードの処理を行う（ステップ S 4 4）。一方、0 x 0 0 が所定回数以上連続した場合（ステップ S 4 5 : Y e s）には、リセット中フラグ R S F がセットされているかどうかを判断する（ステップ S 4 6）。このリセット中フラグ R S F は、ジョブコントロールタスク T 4 0 が管

理するフラグであるので、このステップ S 4 6 の処理は、同一タスク内のフラグを確認する処理と比べて、処理時間が長くなる。

#### 【 0 0 5 6 】

リセット中フラグ R S F がセットされていない場合（ステップ S 4 6 : N o）には、上述したステップ S 4 0 の処理に戻る。一方、リセット中フラグ R S F がセットされている場合（ステップ S 4 6 : Y e s）には、ジョブコントロールタスク T 4 0 に、リセット処理終了確認通知を送信する（ステップ S 4 7）。なぜなら、本実施形態においては、このデータ展開制御処理が、最も上位のループ処理だからである。

#### 【 0 0 5 7 】

次に、メインタスク T 3 0 は、ジョブコントロールタスク T 4 0 からのリセット処理終了確認通知に対する返信を待つ（ステップ S 4 8）。返信があった場合には、このメインタスク T 3 0 内の各種の設定状態を、初期状態に設定する（ステップ S 4 9）。これにより、メインタスク T 3 0 のリセットが完了し、上述したステップ S 4 0 の処理に戻ることになる。

#### 【 0 0 5 8 】

次に、図 6 に示すグラフィックマネージャ処理について説明する。このグラフィックマネージャ処理は、図 5 に示すステップ S 5 0 のグラフィックモードに移行した場合のループである。したがって、このグラフィックマネージャ処理が終了すると、図 5 のステップ S 5 1 の処理に戻ることになる。

#### 【 0 0 5 9 】

図 6 に示すように、メインタスク T 3 0 のグラフィックマネージャ処理では、まず、入力バッファに受信データがあるかどうかを判断する（ステップ S 6 0）。受信データがない場合（ステップ S 6 0 : N o）には、このステップ S 6 0 の処理を繰り返して待機する。

#### 【 0 0 6 0 】

受信データがある場合（ステップ S 6 0 : Y e s）には、入力バッファから 1 バイトのデータを読み込む（ステップ S 6 1）。次に、メインタスク T 3 0 は、この取得したデータが、0 x 0 0 であるかどうかを判断する（ステップ S 6 2）

。取得したデータが、0x00でない場合（ステップS62：No）には、通常のグラフィックモードの処理を行う（ステップS63）。そして、上述したステップS60に戻る。

#### 【0061】

一方、ステップS62で、取得したデータが0x00である場合（ステップS62：Yes）には、ジョブコントロールタスクT40の管理するリセット中フラグRSFが、セットされているかどうかを判断する（ステップS64）。リセット中フラグRSFがセットされていない場合（ステップS64：No）には、上述したステップS60に戻る。

#### 【0062】

一方、リセット中フラグRSFがセットされている場合（ステップS64：Yes）には、グラフィックマネージャ処理から、図5のデータ展開制御処理に戻る（ステップS65）。

#### 【0063】

なお、このグラフィックマネージャ処理では、0x00が所定回数以上連続したかどうかの判断はしていない。これは、グラフィックモードで印刷データとして0x00を受信することは、極めて希であり、0x00を検出するごとにリセット中フラグRSFを確認しに行っても、特にスループットが低下するとは考えられないからである。但し、このような判断をするようにしてもよい。

#### 【0064】

また、本実施形態においては、データ展開制御処理（メインループ処理）の下位のループ処理は、グラフィックマネージャ処理以下にも、数多く存在するが、これらの下位のループ処理にも、ステップS62及びステップS64に相当する処理が、埋め込まれている。また、ループ処理のレベルがさらに深い階層構造になっている場合もある。この場合でも、各ループ処理にステップS62及びステップS64に相当する処理が埋め込まれている。但し、この場合、ステップS65で戻るべきループ処理は、1つだけ上位のループ処理である。したがって、取得したデータが0x00であり、且つ、リセット中フラグRSFがセットされている場合には、各ループ処理は、リセット処理中であると認識して、順次、上位

のループ処理に戻ってきて、最終的に、最も上位のループ処理である図5のデータ展開制御処理に辿り着くことになる。

【0065】

次に、図7及び図8に基づいて、ジョブコントロールタスクT40について説明する。図7は、上述したジョブコントロールタスクT40で行われるジョブコントロール処理を説明するフローチャートであり、図8は、ジョブコントロール処理の下位のループ処理であるネゴシエーション処理を説明するフローチャートである。

【0066】

本実施形態においては、これらのジョブコントロール処理及びネゴシエーション処理は、ROM14に格納されているジョブコントロール処理プログラム及びネゴシエーションプログラムを、リアルタイムOS上でCPU16が実行することにより実現される。

【0067】

図7に示すように、ジョブコントロールタスクT40は、まず、何からのメッセージを受信したかどうかを判断する（ステップS70）。何らメッセージを受信していない場合（ステップS70：No）には、このステップS70の処理を繰り返して待機する。

【0068】

何からのメッセージを受信した場合（ステップS70：Yes）には、リセット中フラグRSFがセットされているかどうかを判断する（ステップS71）。リセット中フラグRSFがセットされていない場合（ステップS71：No）には、受信したメッセージが、制御コマンド解釈タスクT20から送信されたリセットコマンド受信のメッセージであるかどうかを判断する（ステップS72）。

【0069】

受信したメッセージがリセットコマンド受信のメッセージでない場合（ステップS72：No）には、そのメッセージについて必要なジョブを実行し、そのジョブを実行する（ステップS73）。そして、上述したステップS70に戻る。

## 【0070】

一方、上述したステップS72で、受信したメッセージがリセットコマンド受信のメッセージである場合（ステップS72：Yes）には、ジョブコントロールタスクT40は、リセット中フラグRSFをセットする（ステップS74）。続いて、ジョブコントロールタスクT40は、インターフェース制御タスクT10へ、リセット処理移行要求を送信する（ステップS75）。そして、上述したステップS70に戻る。

## 【0071】

一方、上述したステップS71の処理で、リセット中フラグRSFがセットされていると判断した場合（ステップS71：Yes）には、受信したメッセージが、何らかの応答を必要としているメッセージであるかどうかを判断する（ステップS76）。応答が必要なメッセージでない場合（ステップS76：No）には、そのメッセージを読み捨てる（ステップS77）。メッセージを読み捨てることにより、ジョブが読み捨てられたことになる。そして、上述したステップS70の処理に戻る。

## 【0072】

一方、上述したステップS76で、受信したメッセージが応答の必要なメッセージである場合（ステップS76：Yes）には、ジョブコントロールタスクT40は、そのメッセージが、メインタスクT30からのリセット処理終了確認通知であるかどうかを判断する（ステップS78）。受信したメッセージが、リセット処理終了確認通知でない場合（ステップS78：No）には、メインタスクT30がまだ最上位のループ処理に戻ってきていないことになるので、とりあえず、受信したメッセージに対して必要な返事を送信しておく（ステップS79）。これは、メインタスクT30の下位のループ処理で、ジョブコントロールタスクT40からの返信待ちになっているものがある可能性があるため、このループ処理を終了させて上位のループ処理に戻すために、必要な返事のみは返信する必要があるからである。但し、ジョブコントロールタスクT40では、そのメッセージの内容は実行せずに、読み捨てる。そして、上述したステップS70の処理に戻る。

## 【0073】

一方、上述したステップS78で、受信したメッセージが、メインタスクT30からのリセット処理終了確認通知である場合（ステップS78：Yes）には、メインタスクT30が最上位のループに戻ってきたことになるので、リセット中フラグRSFをリセットする（ステップS80）。続いて、ネゴシエーション処理を行う（ステップS81）。なお、このネゴシエーション処理の詳細内容は後述する、

次に、ジョブコントロールタスクT40は、インターフェース制御タスクT10に、リセット処理解除要求を送信する（ステップS82）。続いて、ジョブコントロールタスクT40は、メインタスクT30に、リセット処理終了確認通知の返事を送信する（ステップS83）。そして、上述したステップS70に戻る。

## 【0074】

次に、図8に基づいて、上述した図7のステップS81のネゴシエーション処理について詳しく説明する。

## 【0075】

図8に示すように、まず、ネゴシエーション処理では、プリントエンジン24が動作中であるかどうかを判断する（ステップS90）。プリントエンジン24が動作中である場合（ステップS90：Yes）には、このステップS90の処理を繰り返して、待機する。

## 【0076】

プリントエンジン24が動作中でない場合（ステップS90：No）には、ジョブコントロールタスクT40に残っているタスクが存在するかどうかを判断する（ステップS91）。ジョブコントロールタスクT40に残っているタスクがない場合（ステップS91：No）には、このネゴシエーション処理を終了する。

## 【0077】

一方、ジョブコントロールタスクT40に残っているタスクが存在する場合（ステップS91：Yes）には、そのジョブを捨てて（ステップS92）、メモ

りも解放する（ステップS93）。そして、このネゴシエーション処理を終了する。

#### 【0078】

以上のように、本実施形態に係るマルチファンクションプリンタ10によれば、スキャナ部とプリンタ部とが並行して動作している場合でも、プリンタ部のみをリセットすることができるようにしたので、ユーザの利便性が向上する。すなわち、スキャナ部とプリンタ部とが並行して動作している場合に、プリンタ部をリセットしようとして、スキャナ部のスキャン動作までもが停止してしまう事態を回避することができる。

#### 【0079】

具体的には、プリンタ部リセットコマンドを受信した場合には、インターフェース制御タスクT10が入力バッファにNULLデータを印刷データとして強制的に書き込むようにし、ジョブコントロールタスクT40がリセット処理中であることを示すリセット中フラグRSFをセットすることとした。このため、入力バッファから印刷データを読み出して展開処理を行っているメインタスクT30は、いずれその処理を終えることができる。すなわち、メインタスクT30が階層的なループ処理で構成されている場合、NULLデータに基づいて順次処理を行い、リセット中フラグRSFがセットされていることを検出すると、下位のループ処理から順次抜け出して上位のループ処理に移行するので、最終的には、最上位のメインループ処理に自律的に戻ることができる。

#### 【0080】

このため、複雑なループ処理で構成されている既存のメインタスクT30の処理内容を大きく変更することなく、本実施形態に係るマルチファンクションプリンタ10を実現することができる。

#### 【0081】

なお、本発明は上記実施形態に限定されず種々に変形可能である。例えば、上述した図5のデータ展開制御処理、及び、図6のグラフィックマネージャ処理では、ステップS43及びステップS62で入力バッファから読み込んだデータが0x00であるかどうかの判断をしてから、リセット中フラグRSFがセットさ



れているかどうかを判断したが、これらステップS43及びステップS62の処理を省略して、直接、リセット中フラグRSFがセットされているかどうかを判断するようにしてもよい。

## 【0082】

また、上述した各種のタスクの処理区分は上述したものに限定されるものではなく、複数のタスクの処理をまとめて1つのタスクとして構成してもよく、また、1つのタスクの処理を複数のタスクに分割して構成してもよい。

## 【0083】

さらに、上述した実施形態におけるインターフェース制御タスクT10では、リセット中フラグRSFとは別に、プリンタ部がリセット中であることを示すリセット処理中フラグWSFを設けることとしたが、このリセット処理中フラグWSFを別途設けずにリセット中フラグRSFを共用するようにしてもよい。

## 【0084】

また、上述の実施形態で説明したインターフェース制御タスクT10、制御コマンド解釈タスクT20、メインタスクT30、及び、ジョブコントロールタスクT40等の各タスク処理については、これら各処理を実行するためのプログラムをフロッピーディスク、CD-ROM (Compact Disc-Read Only Memory)、ROM、メモリカード等の記録媒体に記録して、記録媒体の形で頒布することが可能である。この場合、このプログラムが記録された記録媒体をマルチファンクションプリンタ10に読み込ませ、実行させることにより、上述した実施形態を実現することができる。

## 【0085】

また、マルチファンクションプリンタ10は、オペレーティングシステムや別のアプリケーションプログラム等の他のプログラムを備える場合がある。この場合、マルチファンクションプリンタ10の備える他のプログラムを活用し、記録媒体にはそのマルチファンクションプリンタ10が備えるプログラムの中から、上述した実施形態と同等の処理を実現するプログラムを呼び出すような命令を記録するようにしてもよい。

## 【0086】

さらに、このようなプログラムは、記録媒体の形ではなく、ネットワークを通じて搬送波として頒布することも可能である。ネットワーク上を搬送波の形で伝送されたプログラムは、マルチファンクションプリンタ 1 0 に取り込まれて、このプログラムを実行することにより上述した実施形態を実現することができる。

#### 【 0 0 8 7 】

また、記録媒体にプログラムを記録する際や、ネットワーク上を搬送波として伝送される際に、プログラムの暗号化や圧縮化がなされている場合がある。この場合には、これら記録媒体や搬送波からプログラムを読み込んだマルチファンクションプリンタ 1 0 は、そのプログラムの復号化や伸張化を行った上で、実行する必要がある。

#### 【 0 0 8 8 】

##### 【発明の効果】

以上説明したように、本発明によれば、スキャナ部とプリンタ部とを有するマルチファンクションプリンタにおいて、スキャナ部が動作中であっても、プリンタ部だけをリセットすることができるようにしたので、ユーザの利便性が向上する。

##### 【図面の簡単な説明】

#### 【図 1】

本発明の一実施形態に係るマルチファンクションプリンタの内部構成を示すブロック図である。

#### 【図 2】

図 1 のマルチファンクションプリンタがコンピュータからプリンタ部リセットコマンドを受信した場合の処理を概略的に説明するタイミングチャートを示す図である。

#### 【図 3】

本実施形態に係るインターフェース制御処理（インターフェース制御タスク）の内容を説明するフローチャートである。

#### 【図 4】

本実施形態に係る制御コマンド解釈処理（制御コマンド解釈タスク）の内容を

説明するフローチャートである。

【図5】

本実施形態に係るデータ展開制御処理（データ展開制御タスク）の内容を説明するフローチャートである。

【図6】

本実施形態に係るグラフィックマネージャ処理の内容を説明するフローチャートである。

【図7】

本実施形態に係るジョブコントロール処理（ジョブコントロールタスク）の内容を説明するフローチャートである。

【図8】

本実施形態に係るネゴシエーション処理の内容を説明するフローチャートである。

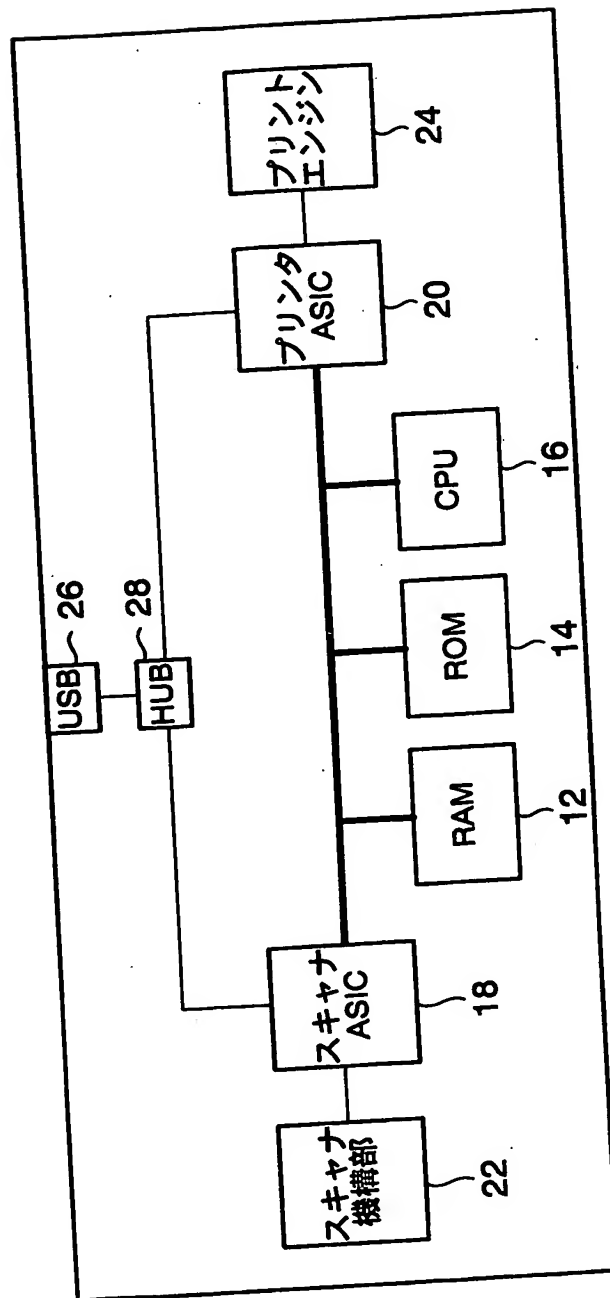
【符号の説明】

- 10 マルチファンクションプリンタ
- 12 RAM
- 14 ROM
- 16 CPU
- 18 スキャナASIC
- 20 プリンタASIC
- 22 スキャナ機構部
- 24 プリントエンジン
- 26 USBポート
- 28 USBハブ
- T10 インターフェース制御タスク
- T20 制御コマンド解釈タスク
- T30 メインタスク
- T40 ジョブコントロールタスク

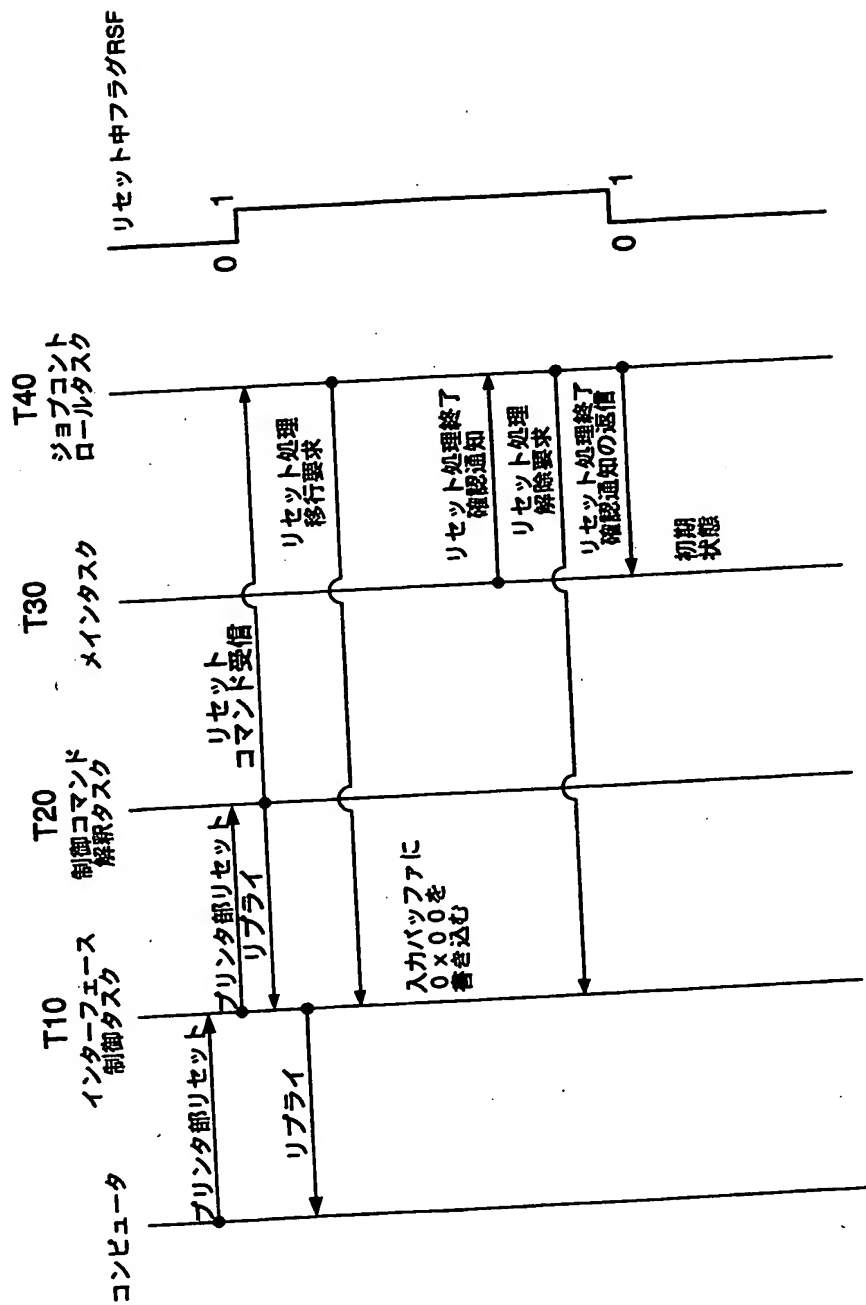
【書類名】 図面

【図1】

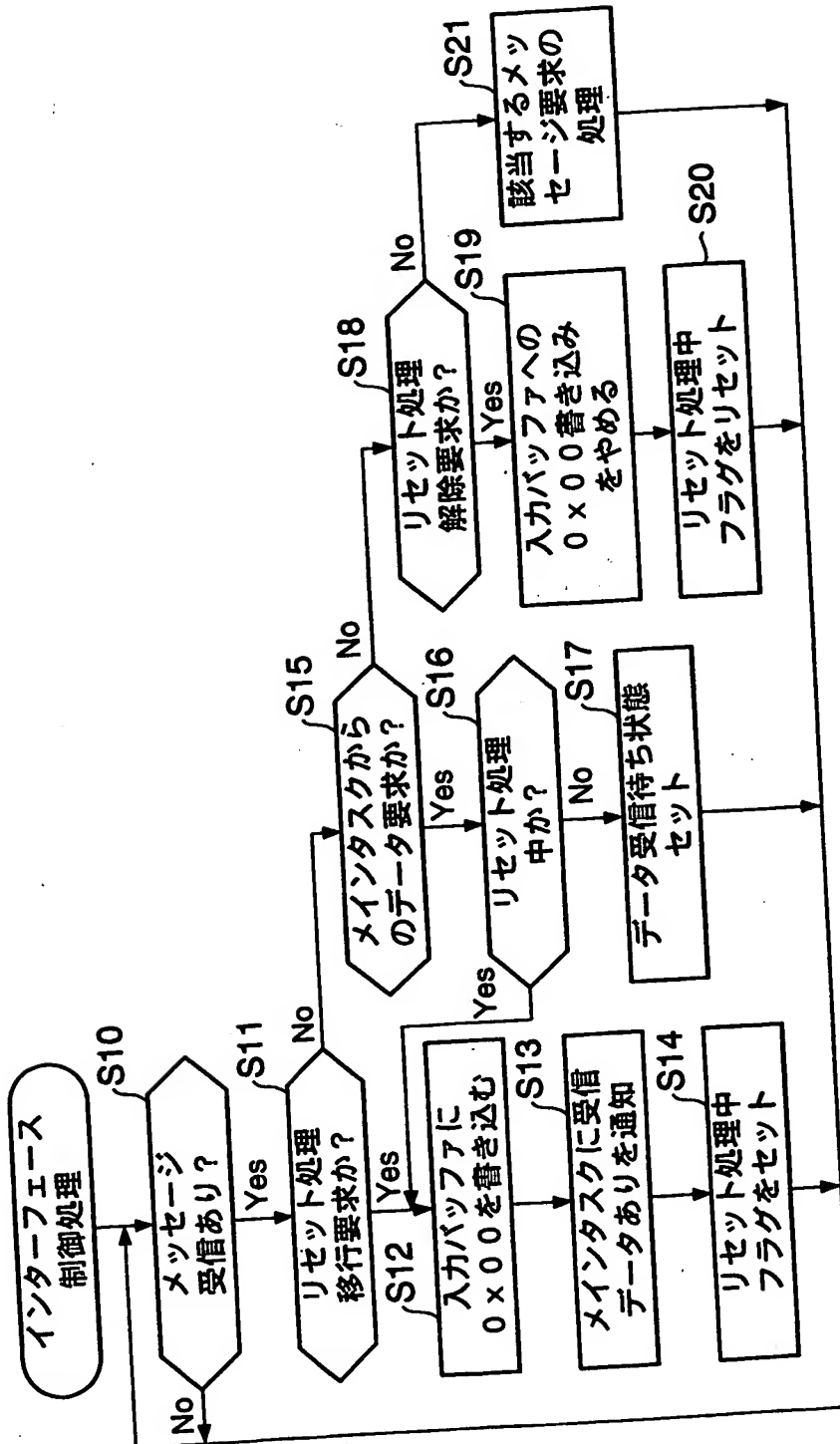
10:マルチファンクションプリンタ



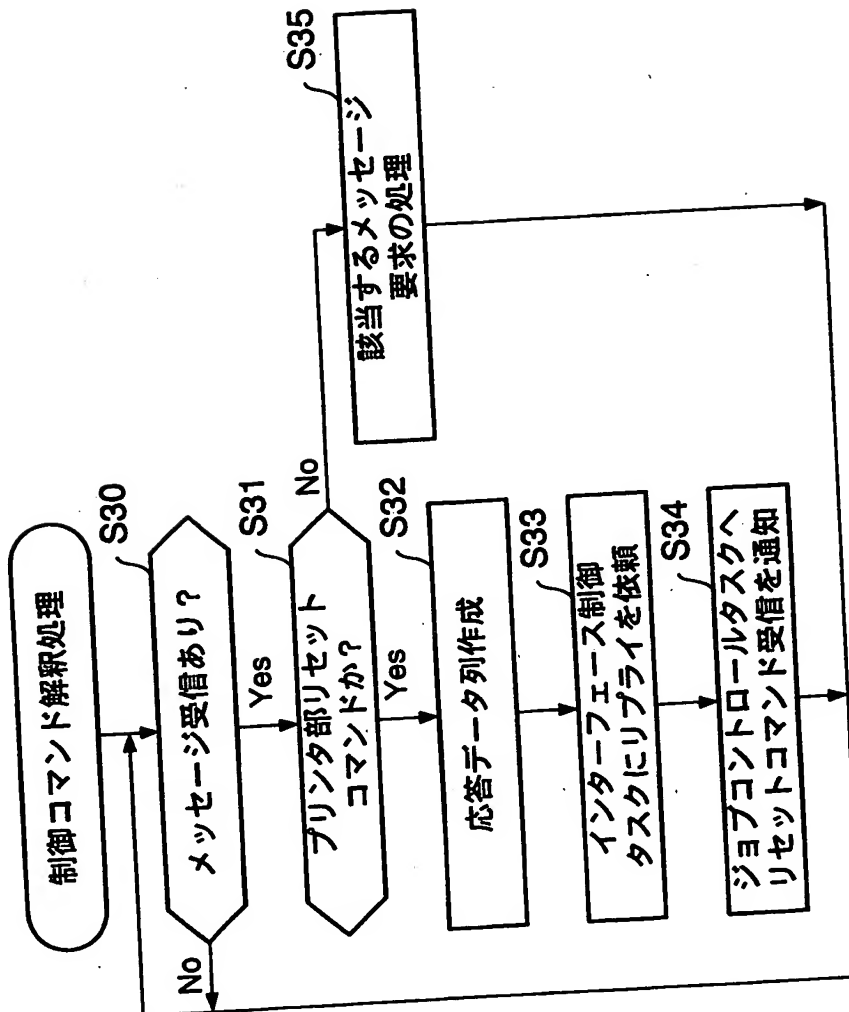
【図2】



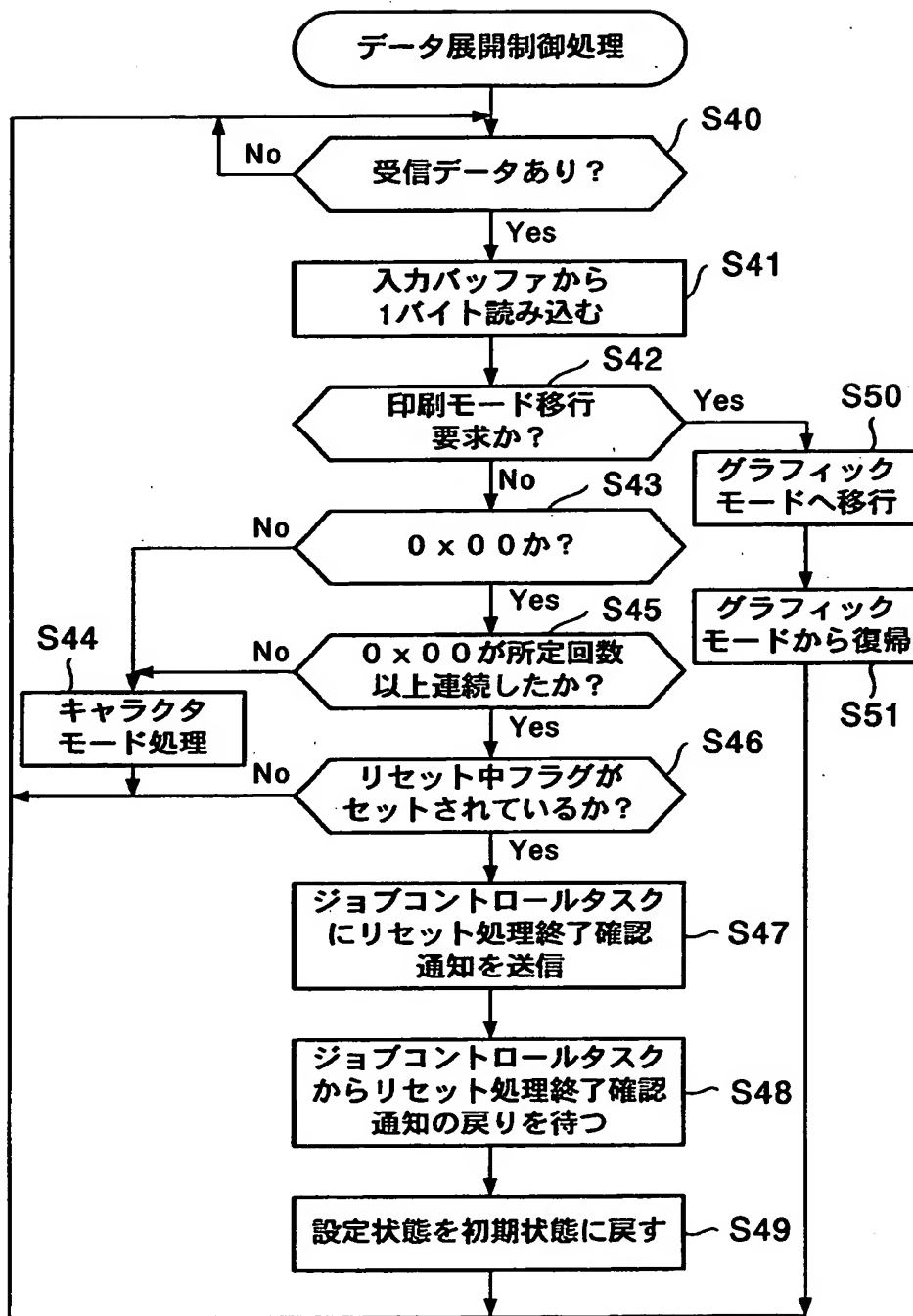
【図 3】



【図 4】

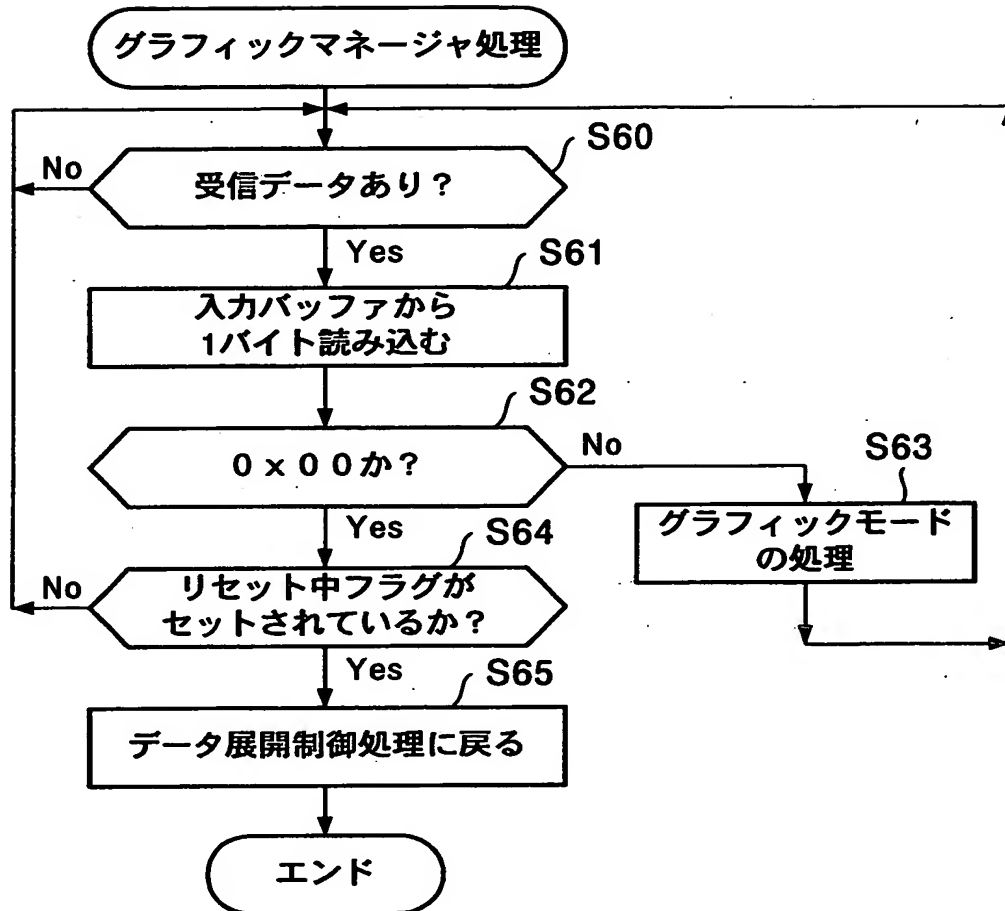


【図 5】

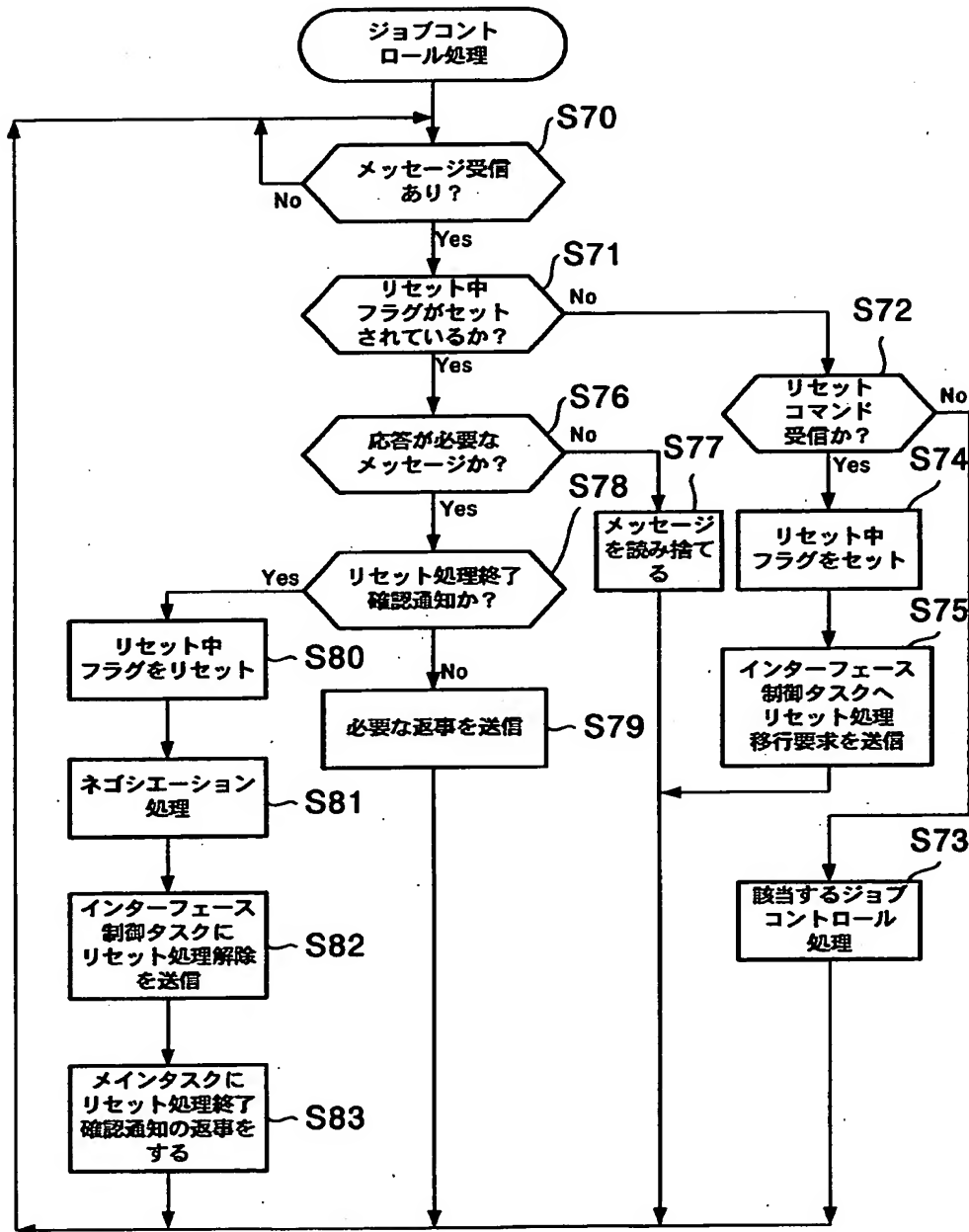




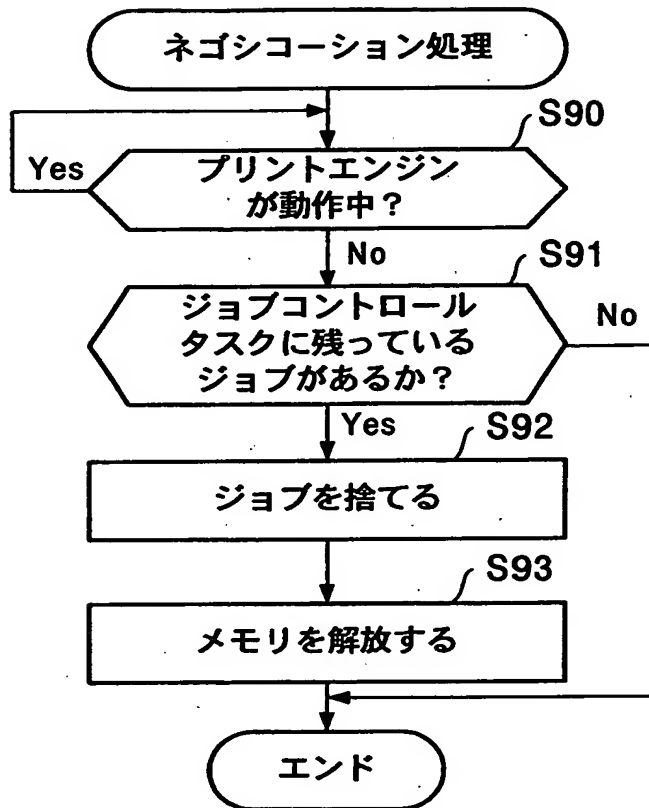
【図 6】



【図 7】



【図 8】



【書類名】 要約書

【要約】

【課題】 スキャナ部とプリンタ部とを有するマルチファンクションプリンタにおいて、プリンタ部のリセットのみを可能にする。

【解決手段】 スキャナとプリンタとが一体化されたマルチファンクションプリンタ10において、プリンタ部リセットコマンドを受信した場合には、インターフェース制御タスクT10は、入力バッファにNULLを強制的に書き込むようにし、ジョブコントロールタスクT40は、リセット中フラグRSFをセットするようにする。メインタスクT30では、階層構造の各種ループ処理が実行されているが、入力バッファにNULLが格納されているのでやがてその処理を終了し、リセット中フラグRSFがセットされているので、メインループ処理に次第に戻ってくる。このため、スキャナ部とプリンタ部とが並行して動作している場合でも、メインタスクT30を初期化でき、プリンタ部のみをリセットすることができるようになる。

【選択図】 図1

特2001-043487

出 願 人 履 歴 情 報

識別番号

[000002369]

1. 変更年月日

1990年 8月20日

[変更理由]

新規登録

住 所

東京都新宿区西新宿2丁目4番1号

氏 名

セイコーエプソン株式会社